**Lezione 27/05/2024 - 2**

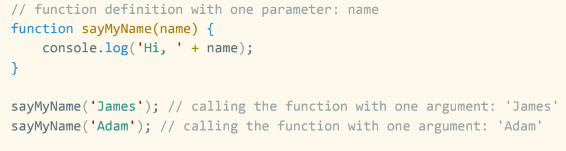
**web developer**

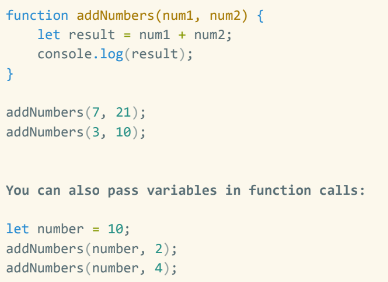
**JavaScript**

**Funzioni:** collezioni riutilizzabili di condizioni / istruzioni /codice

 fatto per Richiamare / Invocare la funzione

le funzioni possono essere impostate con un parametro che può essere datogli per essere eseguito. I parametri possono essere contenuti in variabili





NB:

**Parametri:** variabili preimpostate: *function nome(parametro/i)*

**Argomento:** valore dato direttamente al richiamo della funzione: nomeFunzione(argomento/i)

da considerare anche il **return**: permette di ritornare un valore al termine del lavoro della funzione

se non lo faccio torna **undefined**

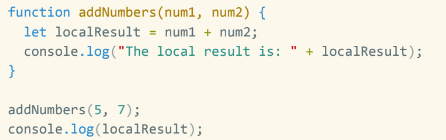
in base alle circostanze le funzioni possono essere usate per espressioni o richiamate dentro altri richiami

**Dipendenza circolare(**tipo di richiamo di funzioni da parte di altre funzioni**)**

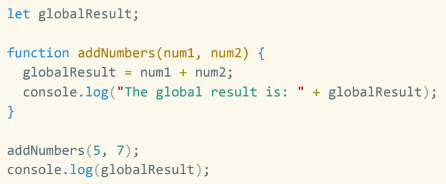
****cicla infinito, riempie spazio stack, pieno = stack overflow

**Portata(Scope) della Variabile:** comportamento della variabile in base (es con let o var, visibili in blocco o funzioni) …

**Locale: dentro funzione, è utilizzabile solo dalla funzione (** dichiarare con **let!**!!**)**

****

**Globale: fuori funzione, è utilizzabile da tutto il codice**

****

**se non si dichiarano le variabili in funzioni JavaScript le imposta come GLOBALI**

**L’indentazione è fondamentale**

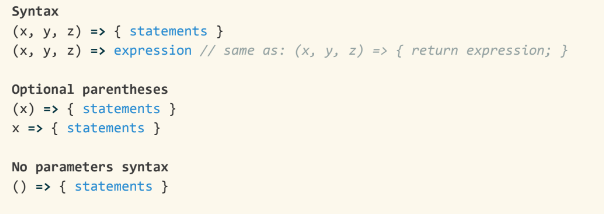
**Let:** confrontato con const, non ha problemi nell’essere usato in funzioni ma

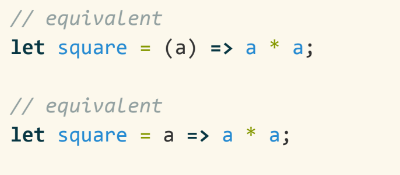
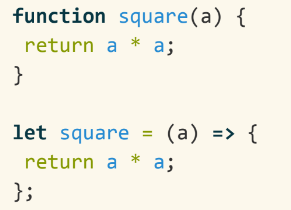
**Var**: è portato per le funzioni e la sua portata è limitata solo nella funzione(era usato nel vecchio JS)

**commenti con // e /\* \*/**

**Function shorthand**

si usa la => ovvero le ***Arrow functions***

******

******

***così i metodi totali per richiamare le funzioni sono…. check quante***